COMMUNITY CONTENT

Home › Submit › 786302

# Submit #786302: code-projects Online Application System For Admission In PHP 1.0 SQL Injection

**Title** code-projects Online Application System For Admission In PHP 1.0 SQL Injection

**Description** A SQL Injection vulnerability exists in the Online Application System for Admission in PHP within the admission form processing functionality.

The vulnerability occurs in the following endpoint:

/OnlineApplicationSystem_PHP/enrollment/admsnform.php

The application processes numerous parameters submitted through an HTTP POST request during the admission process. One of these parameters, detid, is user-controlled and is used by the backend application without proper input validation or sanitization.

Testing confirmed that the detid parameter is vulnerable to time-based SQL injection, indicating that attacker-supplied SQL expressions are interpreted and executed by the database engine.

In the provided request, the attacker injects a delay-based SQL payload using the SLEEP() function:

detid='+(select*from(select(sleep(20)))a)+'

When this request is processed by the application, the server response is delayed by approximately 20 seconds, confirming that the injected SQL query is executed by the database.

This demonstrates that the application directly incorporates user input into SQL queries without using prepared statements or parameterized queries.

Because the parameter is not properly sanitized, attackers can manipulate the SQL query structure and execute arbitrary SQL commands.

**Source** https://github.com/ahmadmarz10-hub/CVEsMarz/blob/main/SQL%20Injection%20in%20Online%20Application%20System%20for%20Admission%20PHP%20detid%20Parameter.md

**User** AhmadMarzouk (UID 95093)

**Submission** 03/23/2026 05:59 PM (14 days ago)

**Moderation** 04/05/2026 10:46 PM (13 days later)

**Status** Accepted

## Community Content

Submissions are made by VulDB community users. VulDB is *not responsible* for their content nor the links to external sources.

Please use the raw information shown and the links listed *with caution*. They might contain malicious and harmful actions, code or data.

The corresponding VulDB entries contain the moderated, verified, and normalized information provided within the raw submission.

## Documentation

- Submission Policy
- Data Processing
- CVE Handling

**VulDB entry** [code-projects Online Application System for Admission 1.0 Endpoint admsnform.php sql injection]

**Points** 20